

Studentská Vědecká Konference 2011

PLÁNOVAČ VLÁKEN SAN

Jiří NOHAVEC¹

1 ÚVOD

Jako odpověď na nedostatky tradičních IP sítí přišla organizace DARPA v roce 1995 s konceptem aktivních sítí. Tento koncept se nedočkal ostrého nasazení z důvodu bezpečnosti a výkonnosti. Vzhledem k teoretické možnosti nalézt řešení na oba problémy, se na KIVu vyvíjí projekt aktivních sítí Smart Active Node (SAN).

V tradičních sítích se mezi jednotlivými uzly přenáší pakety obsahující jen data. V aktivních sítích se v paketech spolu s daty přenáší navíc také programový kód. Tyto pakety se v terminologii aktivních sítí nazývají kapsule. Aktivní síť neposkytuje jen omezenou množinu služeb jako tradiční síť, ale umožňuje svým uživatelům vytvářet a okamžitě nasazovat a testovat své vlastní služby. Chování sítě je možné přizpůsobit dle potřeby.

2 JAVA-IN-JAVA

Projekt SAN je implementace aktivní sítě napsaná v jazyce Java. Uživatelské aplikace a kapsule pro síť SAN jsou také napsány v jazyce Java. Aplikace je spouštěna uživatelem a injektuje (nasazuje) kapsule do sítě. Kapsule putují mezi jednotlivými uzly sítě, které spouštějí jejich kód. To umožňuje kapsulám, aby si sami určovaly svoji cestu (směrování) sítě.

Protože uzel SAN vykonává cizí a potenciálně nebezpečný kód kapsulí, nelze tento kód přímo spouštět v JVM. Je potřeba jej interpretovat. Proto byl pro SAN vytvořen vlastní interpret jazyka Java napsaný v jazyce Java. Tento interpret běží ve standardní JVM.

Pro plánování aplikací a kapsulí nelze použít standardní plánovač JVM. Je potřeba například omezit počet naráz spuštěných kapsulí (zabránění zahlcení) nebo upřednostnit kapsule aplikace vyžadující rychlou odezvu (ping) před kapsulemi vyžadujícími výpočetní výkon (distribuovaný výpočet). Proto bylo potřeba vytvořit vlastní plánovač vláken.

3 FIBER VLÁKNA

Pro zajištění souběžného běhu více aplikací nebo kapsulí je potřeba je spouštět ve vláknech. Nelze použít standardní vlákna JVM, protože neposkytují úplnou kontrolu nad přidělováním procesorového času a nepodporují požadovanou míru zabezpečení pro synchronizační prostředky. Proto bylo nutné vytvořit vlastní implementaci fiber vláken.

Fiber vlákno je vlákno implementované v uživatelském prostoru. Vytváření, spouštění a pozastavování fiber vláken se odehrává zcela bez podpory jádra operačního systému.

Pro zajištění podpory fiber vláken bylo nutné upravit interpret SAN tak, aby byl na požádání schopen pozastavit vykonávanou aplikaci nebo kapsuli a uložit její stav, ze kterého ji bude později možno opět spustit.

¹ Jiří Nohavec, student navazujícího studijního programu Inženýrská informatika, obor Softwarové inženýrství, e-mail: jnohavec@students.zcu.cz

Výhodou fiber vláken je možnost spustit více instancí interpretrů a plně využít potenciál víceprocesorových konfigurací. Je však potřeba plánovač, který bude aplikacím a kapsulím přidělovat a odebírat interpretr.

4 PLÁNOVAČ

Plánovač je komponenta uzlu SAN, která rozhoduje o přidělování procesorového času aplikacím a kapsulím. Dále se plánovač stará o zajištění synchronizace.

Z hlediska architektury lze plánovač rozdělit na následující části:

- Plánovací strategie – Vybírá aplikaci nebo kapsuli, které bude na následující časové kvantum přidělen procesor (interpretr). V současné době je implementováno několik různých algoritmů – prioritní plánování, Round Robin, plánování loterií, ...
- Optimalizace pro více procesorů – Pokud je k dispozici více procesorů, může se stát, že plánovač bude jednu aplikaci plánovat střídavě na různé procesory a nevyužije se tak potenciál vyrovnávacích cache pamětí. Proto je nutno zajistit mezivrstvu mezi plánovacím algoritmem a zavaděčem, která se postará o optimální přiřazení procesorů běžícím aplikacím a kapsulím.
- Zavaděč – Součást plánovače, která řídí spouštění a pozastavování aplikací a kapsulí podle rozhodnutí plánovací strategie.
- Modul pro správu podmínkových proměnných.
- Modul pro správu synchronizačních monitorů.
- Modul pro detekci uvíznutí – Sestavuje graf alokace zdrojů (RAG) a pomocí prohledávání do šířky (BFS) vyhledá smyčky. Smyčka v RAG znamená uvíznutí. Uvíznutí je řešeno násilným ukončením jedné aplikace nebo kapsule v cyklu.
- Modul pro prevenci uvíznutí – Využívá bankéřova algoritmu pro předvídání uvíznutí. Vyžaduje spolupráci aplikací, které mohou dopředu oznámit, které monitory a v jakém pořadí budou zabírat.

4 DOSAŽENÉ VÝSLEDKY

Byla vytvořena vlastní implementace fiber vláken a plánovač s dostatečnou podporou zabezpečení. Bezpečnosti bylo dosaženo na úkor výkonu. Režie na provoz fiber vláken zpomalila výkon interpretace.

5 ZÁVĚR

Do budoucna se počítá s urychlením interpretace pomocí přímého překladu bytecode interpretovaných aplikací a kapsulí do instrukční sady procesoru (podobně jako pracuje JIT v JVM). Projekt ponese jméno SAN++.

LITERATURA

Koutný, Tomáš, 2005. *Reference Implementation of Grade32 AN Server*

Rejda, Michal, 2008. Diplomová práce – *Server aktivní sítě*, FAV ZČU

Syrovátka, Jan, 2009. Diplomová práce – *Interpretace kódu v Aktivních sítích*, FAV ZČU